

Écrire du C - Code::Blocks IDE

Code::Block est un environnement de développement C/C++ gratuit et open source. Il inclut le compilateur gratuit MinGW, compatible avec les outils GCC ('GNU Compiler Collection', généralement rencontrés sous Linux...)

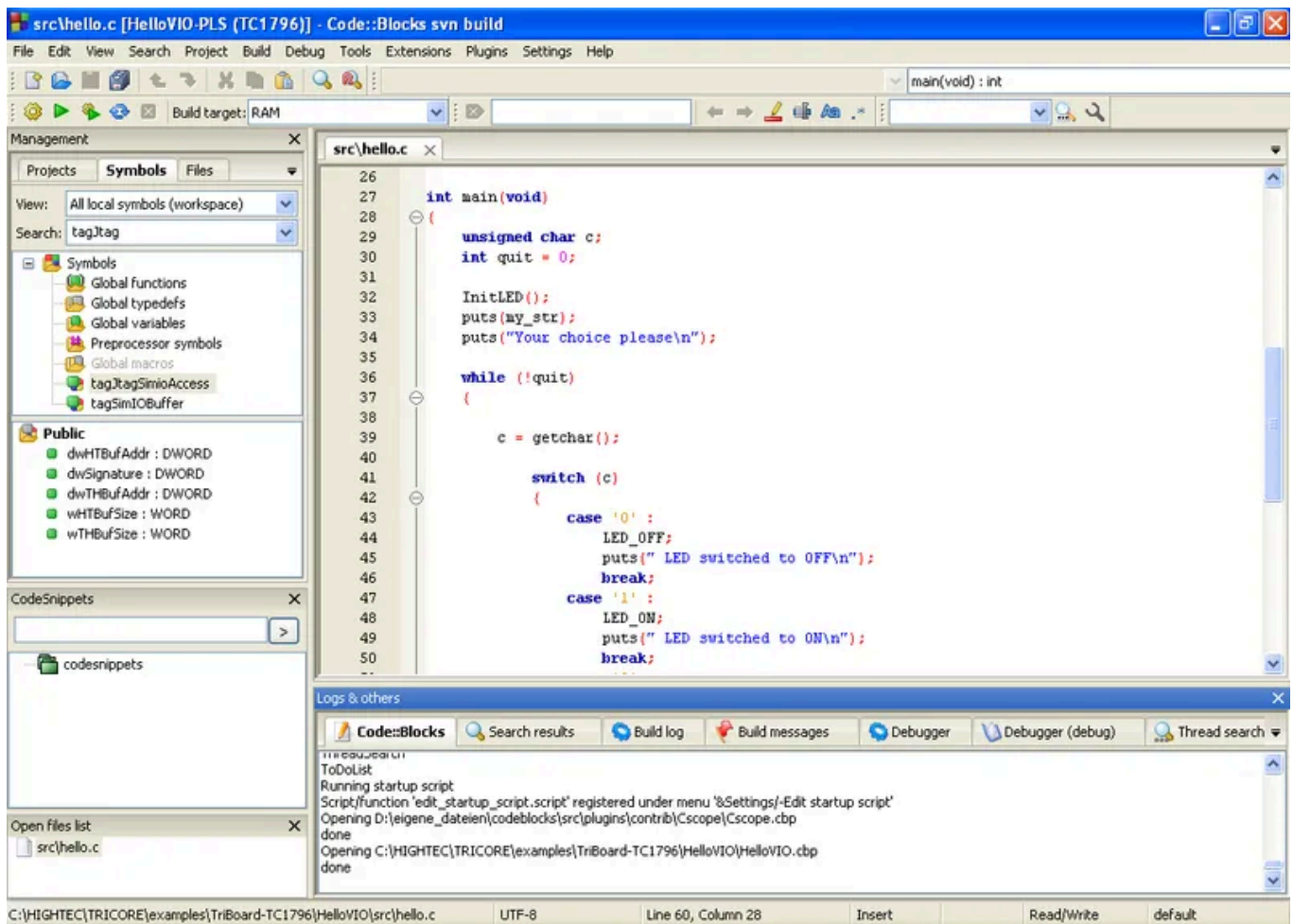
5TTR

6TTR



Découverte

Installer et utiliser Code::Blocks



Objectifs

À la fin de ce chapitre, tu sauras :

- Installer Code::Blocks avec le compilateur inclus.
- Créer ton premier projet en C.
- Compiler et exécuter un programme.
- Comprendre l'organisation de l'interface.

- Éviter les erreurs fréquentes au démarrage.

Pourquoi Code::Blocks ?

Code::Blocks est un **environnement de développement (IDE)** spécialement conçu pour les langages compilés comme C et C++.

- Il regroupe tout ce dont tu as besoin : **éditeur de code, compilateur, outil pour exécuter tes programmes.**
- Avantage : tu n'as pas besoin de configurer séparément un compilateur.

Installation

🌟 Pense bien à télécharger la version **AVEC LE COMPILATEUR MINGW**.

1. Va sur le site officiel : <https://www.codeblocks.org/downloads/binaries/>.
2. Choisis la version **Code::Blocks with MinGW** (c'est celle qui contient déjà le compilateur `gcc`): exemple: `codeblocks-25.03mingw-setup.exe`.
 - ⚠ Ne télécharge pas la version "sans MinGW", sinon il te manquera le compilateur.
3. Lance l'installation (clic sur "Suivant" plusieurs fois).
4. Laisse les options par défaut.
5. À la fin, ouvre Code::Blocks pour vérifier que tout fonctionne.

Tu peux télécharger la version installable, auquel tu devras demander un accès admin au moment de l'installation, ou la version "nosetup" (portable) auquel cas tu devras créer les raccourcis toi-même.

Première utilisation

Quand tu ouvres Code::Blocks :

- Barre de menus en haut (Fichier, Éditer, Projet...).
- Zone principale pour écrire ton code.
- Panneau de messages en bas (erreurs, avertissements).
- Boutons rapides (nouveau, ouvrir, compiler, exécuter).

Créer ton premier projet

1. Clique sur **File > New > Project**.
2. Choisis **Console Application** → **Next**.
3. Sélectionne **C language** → **Next**.
4. Donne un **nom à ton projet** (ex: `HelloWorld`).
5. Choisis un dossier où sauvegarder.
6. Clique sur **Finish**.

Code::Blocks génère automatiquement un fichier `main.c` avec un programme exemple.

Compiler et exécuter

- Pour **compiler** ton programme : clique sur le bouton en forme de **roue dentée**.
- Pour **lancer** ton programme : clique sur le bouton **triangle vert**.
- Pour faire les deux en même temps : clique sur **Build & Run (F9)**.

👉 Si tout va bien, une fenêtre noire (console) s'ouvre et affiche :

```
Hello world!
```

Erreurs fréquentes

- Si Code::Blocks ne trouve pas le compilateur :
 - Vérifie que tu as bien téléchargé la version *with MinGW*.
- Fenêtre console qui se ferme trop vite :
 - Ajoute une ligne `getchar();` avant `return 0;` dans `main.c` .

Exercices

1. Installe Code::Blocks sur ton ordinateur personnel.
2. Crée un projet console nommé `Test` .
3. Modifie le fichier `main.c` pour afficher ton prénom.

4. Ajoute une deuxième ligne qui affiche ton âge.
5. Compile et exécute ton programme.

À retenir

- Télécharge toujours la version **Code::Blocks with MinGW**.
- Un projet console est le type de projet à utiliser pour apprendre le C.
- La compilation est indispensable avant l'exécution.
- Les erreurs affichées en bas de l'écran sont **précieuses** pour corriger ton code.

Notes pour l'enseignant

Objectifs pédagogiques

- Permettre à chaque élève d'avoir un environnement de développement fonctionnel.
- Introduire la notion de **projet** et d'organisation des fichiers.
- Dédramatiser les premiers messages d'erreurs de compilation.

Points-clés à insister

- Toujours choisir la version avec MinGW (évite 90 % des problèmes débutants).
- Insister sur le rôle du bouton **Build & Run** (F9).
- Les erreurs de compilation sont normales, il faut les lire attentivement.

Bonnes pratiques

- Demander aux élèves de **nommer clairement** leurs projets (pas `projet1`, mais `HelloWorld`, `Calculatrice` ...).
- Leur faire tester volontairement une erreur (oublier un `;`) pour voir le message affiché.
- Montrer que Code::Blocks est un **outil de travail**, pas seulement une "boîte magique".